Fachhochschule Köln
Cologne University of Applied Sciences

Institut für Medien- und Phototechnik

Bachelorarbeit

Entwicklung eines dialogorientierten Charakters mit Sprachsteuerung auf mobilen Endgeräten

vorgelegt von

Rouven von der Burg

Mat.-Nr. 11068765

Erstgutachter: Prof. Dr. Stefan Grünvogel (Fachhochschule Köln)

Zweitgutachter: Dipl-Wirt.Ing. Jens Piesk (Nurogames GmbH, Köln)

Juni 2012

Fachhochschule Köln
Cologne University of Applied Sciences

Institut für Medien- und Phototechnik

Bachelor Thesis

Development of a dialogue-orientated character with voice control for mobile devices

submitted by

Rouven von der Burg

Mat.-Nr. 11068765

First Reviewer: Prof. Dr. Stefan Grünvogel (Cologne University of Applied Sciences)

Second Reviewer: Dipl-Wirt.Ing. Jens Piesk (Nurogames GmbH, Cologne)

June 2012

Bachelorarbeit

Titel: Entwicklung eines dialogorientierten Charakters mit Sprachsteuerung auf mobilen Endgeräten

Autor: Rouven von der Burg

Gutachter:

- Prof. Dr. Stefan Grünvogel (Fachhochschule Köln)
- Dipl-Wirt.Ing. Jens Piesk (Nurogames GmbH, Köln)

Zusammenfassung: Diese Bachelorarbeit erläutert das Vorgehen bei der Erstellung einer Applikation für mobile Endgeräte, die einen dialogorientierten 3D-Charakter darstellt, der auf die Eingaben des Benutzers reagiert. Erläutert wird hierbei auch die historische Entwicklung der Mensch-Maschinen-Kommunikation. Darauf folgt ein Überblick über die, im Bereich der mobilen Endgeräte verfügbare, Hard- und Software sowie deren Architektur. Weitere Teile befassen sich mit den in der Applikation genutzten, selbst erstellten Methoden zur Textanalyse und Antwortgenerierung, sowie der Kombination aller Komponenten zu einem Endprodukt.

Stichwörter: Sprachsteuerung, mobile Endgeräte, Android, 3D-Charakter, Eliza

Datum: 28. Juni 2012

Bachelors Thesis

Title: Development of a dialogue-orientated character with voice control for mobile devices

Author: Rouven von der Burg

Reviewers:

- Prof. Dr. Stefan Grünvogel (Cologne University of Applied Sciences)
- Dipl-Wirt.Ing. Jens Piesk (Nurogames GmbH, Cologne)

Abstract: Abstract: The bachelor thesis at hand illustrates the procedure of generating an application for mobile devices. This application is of a dialogue-orientated 3D-character and reacts to the input of the users. In addition, the historical development of humans-machine-communication is illustrated. After that a survey of hard- and software available for the mobile devices as well as their architecture is given. Further parts of the thesis deal with the self-generated methods of text analysis and answer-generation as well as the combination of these components to create an end product.

Keywords: voice control, mobile devices, Android, 3D-Character, Eliza

Date: 28 June 2012

Inhalt

1.Einleitung

Die Idee dieser Bachelorarbeit ist es, das von *Joseph Weizenbaum* bereits 1966 entwickelte Eliza Programm aufzugreifen, und auf die heute allgegenwärtigen Smartphones zu adaptieren, sowie dieses um neue technische Aspekte zu erweitern.

Bei dem Eliza Programm handelt es sich um eine Software, die auf die Eingaben des Benutzers reagiert, und so versucht eine zwischenmenschliche Kommunikation zu simulieren.

Ziel dieser Arbeit ist die Erstellung einer Applikation (App) für ein aktuelles Smartphone Betriebssystem. Diese soll die, nun auch auf Smartphones verfügbaren, neuen Elemente wie Spracherkennung und Sprachsynthese nutzen, um einen Mehrwert gegenüber der ursprünglichen Software zu erstellen.

Dazu soll STT (Speech-to-Text) genutzt werden. Der Benutzer spricht hierbei mit seinem Smartphone und erhält dann eine für den Inhalt der Eingangsnachricht passende Antwort. Diese wird dann wiederum, auf Basis des TTS (Text-to-Speech), in eine hörbare Sprache umgewandelt.

Genutzt werden sollen hierbei bereits vorhandene Schnittstellen für die Bereiche STT und TTS. Diese werden mit eigenen Softwarefragmenten, wie der Inhaltsanalyse und der optischen Darstellung, verknüpft, um so ein in sich stimmiges Gesamtprojekt zu schaffen.

Ein weiteres Ziel dieser Arbeit befasst sich mit dem Erstellen entsprechender Wortdatenbanken, um die Applikation mit entsprechenden Wörtern sowie Satzfragmenten auszustatten.

Um die Interaktion mit dem Benutzer ansprechender zu gestalten, wird während der Konversation ein Charakter dargestellt, der auf die Spracheingaben des Benutzers reagiert.

Aufgegriffen werden soll hierbei auch die Idee der *Klientenzentrierten Psychotherapie.*

Bei der *Klientenzentierten Psychotherapie* handelt es sich um eine von *Carl R. Rogers* entwickelte Art der Gesprächspsychotherapie. Bei dieser Art der Psychotherapie rückt die zu therapierende Person in den Gesprächsmittelpunkt. Das heißt, es werden primär die Wünsche, Gedanken und Gefühle des zu Therapierenden beachtet. Dabei wird von Seiten des Therapeuten keine Beurteilung des zu Therapierenden vorgenommen und es erfolgen auch keine Ratschläge (nicht-direktives Verhalten). Der Therapeut dient einzig und allein als Reflektor des Gesagten und hält durch sein Wiederaufgreifen der vom Patienten angebotenen Inhalte das Gespräch in Gang. Dadurch, dass er ansonsten nicht aktiv in den Gesprächsverlauf eingreift, kann sich das Gespräch, frei nach der vom Patienten, teils unterbewusst, teils gewünschten Richtung entwickeln. (Rog93)

Die Applikation soll natürlich nicht den Therapeuten an sich ersetzen, kann aber durch die Möglichkeit *„des sich etwas von der Seele Redens“* dem Benutzer, bei alltäglichen Problemen, eine gewisse seelische Erleichterung schaffen.

Eine weitere Möglichkeit der Anwendung ist der reine Unterhaltungszweck für den Benutzer. So kann dieser die Grenzen der Anwendung austesten oder sich einfach mit diesem Charakter entspannt unterhalten.

Die während der Bachelorarbeit erstellte Applikation wird bei der Softwarefirma *Nurogames* von mir entwickelt und soll dort im Rahmen der weiteren Entwicklung von Applikationen als Grundlage genutzt werden.

Ein weiteres Ziel ist die spätere Veröffentlichung der Applikation im zum Betriebssystem zugehörigen Store.

1.1 Aufbau der Arbeit

Zu Beginn der Arbeit wird in Kapitel 2 eine Zusammenfassung gegeben was das Eliza Programm überhaupt ist und wie es sich über die Zeit entwickelt hat. Dabei werden auch Weiterentwicklungen des Ursprungsprogrammes dargestellt. Des Weiteren wird auf den Turing-Test eingegangen.

In Kapitel 3 wird dargelegt, welche Grundvoraussetzungen an die Hard- sowie Software gestellt werden. Dies trägt dazu bei, die Auswahl aus einem der drei aktuellen Betriebssysteme für mobile Endgeräte (iOS, Android, Windows Phone 7) zu begründen.

In Kapitel 4 wird die Wahl des Android Betriebssystems begründet und die Grundeigenschaften dieses Systems erklärt. Dabei wird sowohl auf die Hard- sowie auch auf die Software eingegangen.

Das Kapitel 5 befasst sich mit der Softwareentwicklung der Applikation, also der Konzeptionierung, der Architektur, sowie der Nutzung externer Bibliotheken. In diesem Kapitel wird auch ein Überblick über die für eine Android Applikation übliche Struktur, sowie deren Einschränkungen gegeben. Des Weiteren werden die selbst entwickelten Softwareelemente und deren Funktion erläutert. Dabei wird vor allem auf die Antworterstellung eingegangen.

Das Kapitel 6 befasst sich mit dem Charakter Design und dessen Einfluss auf die Applikation.

2.Das Eliza Programm

Entwickelt wurde das Eliza Programm von *Joseph Weizenbaum.* Bei diesem handelt es sich um einen, im Jahre 1923 in Berlin geborenen, Informatiker, Gesellschafts- und Wissenschaftskritiker. Er befasste sich unter anderem mit den Gebieten der künstlichen Intelligenz, sowie mit dem kritischen Umgang mit dem Gebiet der Informationstechnologie.(Wei78)

2.1 Ursprung des Eliza Programms

Bei dem Eliza Programm handelt es sich um eine Entwicklung von *Joseph Weizenbaum* aus dem Jahr 1966. Geschrieben wurde das Programm in MAD-SLIP für einen IMB 7094. Zweck des Programms war es, die Kommunikation zwischen einem Computer und einer realen Person zu simulieren. Hierbei schrieb der Benutzer des Programms Sätze in die Eingabekonsole und erhielt dann im Gegenzug von diesem eine Antwort. Diese Antwort, abhängig vom Umfang der vorhandenen Skripte und Datenbanken, gestaltete sich inhaltlich nach den vorherigen Eingaben des Benutzers.

Eine Art der Antwort des Eliza Programms war es, auf eine vorherige Eingabe mit einer Frage zu reagieren. Dabei werden Nomen oder Verben aus der vorherigen Eingabe genutzt und dann in diese Frage an der passenden Stelle eingebaut.

Eine Antwort auf einen einfachen Eingabesatz konnte zum Beispiel so aussehen:

Benutzer : „Ich hatte heute Stress in der Uni.“

Eliza : „Warum hattest du heute Stress in der Uni?“

Hierbei wurden im Eingabesatz erkannte Wörter wiederverwendet, um im Anschluss daran einen passenden Antwortsatz zu generieren. Dies funktioniert soweit verlässlich, solang der Anwender nur die in der Datenbank verwendeten Schlüsselwörter benutzt und somit im angedachten Themengebiet bleibt.

Eine andere Art der Antwort war die Reaktion auf gewisse Schlüsselwörter, die dann zu einer fest vorgegebenen Antwort führten.

Benutzer : „Auf Wiedersehen.“

Eliza : „Vielen Dank für das nette Gespräch.“

Dieses System konnte, solange man sich in einem gewissen Themenbereich befand, eine halbwegs reale Konversation simulieren. Wurde aber zum Beispiel ein Schlüsselwort „falsch“ interpretiert konnte das Gespräch in eine falsche und unpassende Richtung gelenkt werden.

Benutzer : „Ich habe einen ziemlichen Kater.“

Eliza : „Erzähl mir mehr über dein Haustier.“

2.2 Weiterentwicklungen des Eliza Programms

Aus der ursprünglichen Version des Eliza Programms, in seiner rein textbasierenden Form, haben sich bis heute viele verschiedene Versionen mit unterschiedlichen Benutzeroberflächen entwickelt.

Bei der Weiterentwicklung der Software wurden einerseits optische Verbesserungen am Programm vorgenommen, andererseits wurde auch die Textanalyse schrittweise verbessert.

Angelehnt an das Eliza Programm gibt es im Internet einige Chatbots, die genau wie das ursprüngliche Programm, eine reale Kommunikation zwischen dem Benutzer und der Homepage simulieren sollen. Dieses System wird von verschiedenen Firmen genutzt, um direkt auf ihrer Homepage einen virtuellen Ansprechpartner für ihre Kunden zur Verfügung zu stellen.

2.3 Der Touring-Test

Beim Turing-Test handelt es sich um einen, von *Alan Turing* entwickelten, Versuch aus dem Jahre 1950. Ziel des Testes war es zu zeigen, dass ein Computer ein ähnliches Denkvermögen wie ein Mensch aufweisen kann.

Bei *Alan Turing* handelt es sich, um einen in London geborenen Mathematiker, Logiker und Kryptoanalytiker. Vor allem durch seine Beiträge im Bereich der theoretischen Informatik und die Entwicklung der Turingmaschine gilt er heutzutage als eine der Personen, die den Grundstein für den modernen Computer gelegt haben. Insbesondere befasste er sich mit dem Gebiet der künstlichen Intelligenz. Dies führte unter anderem auch zur Entwicklung des Turing-Testes.

Beim, in Abbildung 1 dargestellten, Turing-Test kommuniziert eine Testperson (C) rein über eine Tastatur und einen Bildschirm mit zwei anderen Gesprächspartnern. Dabei handelt es sich bei dem einem der Gesprächspartner um eine Maschine (B) und bei dem anderen um einen Menschen (A). Im Gesprächsverlauf sollen nun die beiden Gesprächspartner die Testperson mit ihren Antworten davon überzeugen, dass sie die denkenden Menschen und keine Maschine sind. Sollte die Testperson nach einem längeren Gespräch mit beiden keine klare Aussage treffen können wer die Maschine ist, so ist der Turing-Test bestanden. (ERB09)

Beim Bestehen des Testes wird der Maschine ein gleichwertiges Denkvermögen ähnlich dem des Menschen unterstellt.

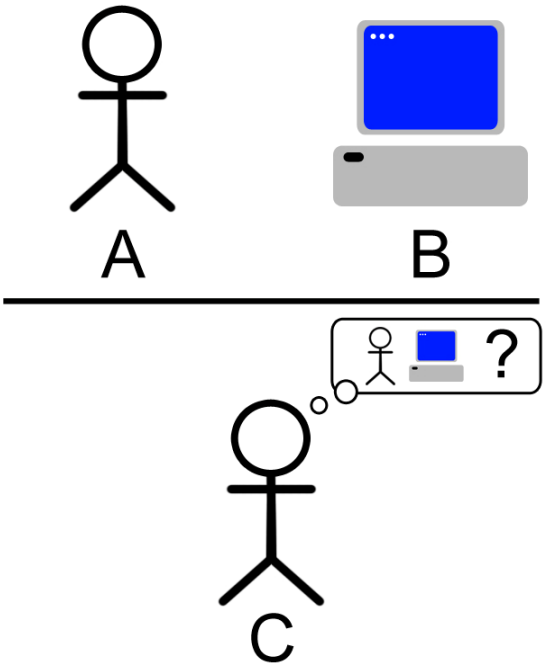

Abbildung 1 Darstellung des Turing-Testes

Turing war davon überzeugt, dass es bis zum Jahr 2000 entsprechend leistungsfähige Computer und Software gäbe, bei der die Unterscheidung zwischen Mensch und Maschine, nach einer fünfminütigen Konversation, bei maximal 30 Prozent läge.

3.Allgemeine Anforderungen

Im Folgenden wird die Auswahl des passenden Betriebssystems und der dazugehörigen Hardware für die Erstellung der Applikation erläutert. Vor allem die nachfolgenden Faktoren sind entscheidungsgebend für die Auswahl des Betriebssystems.

Hierbei muss nicht nur das reine Betriebssystem auf seine Tauglichkeit überprüft werden, sondern auch die Hardware der jeweiligen Geräte die dieses nutzen.

Beachtet werden hierbei nur reine Smartphone Betriebssysteme, da ältere Systeme keinesfalls die benötigten Spezifikationen aufweisen können.

3.1 Display

Grundvoraussetzung, um die Applikation für ein Betriebssystem zu erstellen, sind hochauflösende Displays. Die derzeit auf dem Markt befindlichen Geräte haben Displays mit Auflösungen von mindestens 240x320 Pixeln, die für die Erstellung der Applikation eine ausreichende Qualität liefern. Diese Displays reichen aus, um die Userinterface Elemente und auch den animierten Charakter in der gewünschten Qualität darzustellen.

3.2 Leistung

Die Leistungsfähigkeit des Betriebssystems sowie der genutzten Hardware muss ausreichend sein um eine flüssige Darstellung der Animation zu gewährleisten. Diese muss ebenso reibungslos funktionieren, wenn gleichzeitig Prozesse wie die Textanalyse, Antworterstellung oder auch der Datentransfer im Hintergrund ablaufen.

Da die verbauten Prozessoren in den Geräten der letzten Generationen jedoch alle einen Prozessor mit mindestens einem Gigahertz besitzen, kann an dieser Stelle auch noch kein Ausschluss eines Systems getroffen werden.

Des Weiteren müssen die Geräte mindestens über den EDGE Standard der Internetverbindung verfügen, um den anfallenden Datenaustausch entsprechenden schnell zu erledigen.

So kann also von Seiten der Hardware noch keines der aktuellen Betriebssysteme im Bereich der Smartphones ausgeschlossen werden.

3.3 Vorhandene Bibliotheken

Besondere Rücksicht muss auch auf bereits vorhandene Bibliotheken genommen werden. Essenziel sind für diese Applikation Bibliotheken die den Bereich Speech-to-Text sowie Text-to-Speech abdecken, da ohne diese das Erstellen der Applikation nicht möglich ist bzw. diese komplexen Elemente (STT und TTS) selbst erstellt werden müssten.

3.4 Verbreitung

Da die Applikation auch nach ihrer Fertigstellung im jeweils zugehörigen Store verkauft werden soll, muss das Betriebssystem einen ausreichenden Marktanteil haben, um so eine relevante Zielgruppe erreichen zu können.

4.Wahl des Betriebssystems

Beim Betrachten der momentan verfügbaren Betriebssysteme mit einem entsprechenden Marktanteil, welche die in Kapitel 3 angedachte technische Umsetzung ermöglichen, konnte die Auswahl auf drei Betriebssysteme für mobile Endgeräte reduziert werden:

Apple’s iOS

Google’s Android

Microsoft’s Windows Mobile 7

Abbildung 2 Logos der Betriebssysteme

Meine Wahl viel hierbei auf das von Google erstelle Android Betriebssystem. Dieses ist offener gestaltet als das iOS von Apple und verbreiteter und weiter entwickelt als das Windows Mobile 7 von Microsoft.

So können zum Beispiel Applikationen, die von Privatpersonen oder kleineren Firmen hergestellt werden, auch ohne das Durchlaufen einer Testphase der Herstellerfirma auf Geräten mit dem Betriebssystem installiert werden. Wohingegen dies beim iOS ohne eine, vom Hersteller nicht erlaubte, Modifikation des Betriebssystems nicht möglich ist.

Google Android bietet weiter den Vorteil, dass für dieses System erstellte Applikationen nicht nur auf Smartphones genutzt werden können, sondern auch auf Desktop Computern mit installierter Android Software.

Ferner können auch Programmteile aus einer Vielzahl vorhandener Java Bibliotheken genutzt werden, da die Entwicklung von Applikationen für Android auf Java und C/C++ beruht. Dies bietet die Möglichkeit, den im Laufe der Bachelorarbeit erstellten Quellcode für spätere Projekte auch auf anderen Plattformen als Smartphones bzw. dem Android Betriebssystem zu nutzen.

So konnten auch von mir bereits im vorrausgegangen Praktikum sowie Studium gesammelte Erfahrungen, zum Beispiel im Bereich der Java Programmierung, genutzt und auf dieses Projekt übertragen werden.

4.1 Grundlagen des Android Betriebssystems

Bei Android handelt es sich um ein Betriebssystem für Smartphones, Tablets sowie Netbooks das seit 2005 von Google und anderen Mitgliedern der Open Handset Alliace (OHA) weiter entwickelt wird. Basis dieser Software ist der Linux-Kernel 2.6 . Grundidee des Android Systems ist der Gedanke der freien Software und Quelloffenheit.

Google gab erst am 5. November 2007 bekannt, dass sie dieses zusammen mit der OHA entwickeln und veröffentlichten bereits am 21. Oktober 2008 die erste Version des Betriebssystems.

Alle darauf Folgenden Versionen sind in alphabetischer Reihenfolge nach Süßspeisen benannt.

Versionsnummer	Name
1.1	-
1.5	Cupcake
1.6	Donut
2.0/2.1	Eclair
2.2.x	Froyo
2.3.x	Gingerbread
3.x (Tablets Only)	Honeycomb
4.0	Ice Cream Sandwich

Tabelle 1 Versionen des Android Betriebssystems

Die momentan aktuellste Version von Android ist „Ice Cream Sandwich“ (Version 4.0).

4.2 Technische Details

4.2.1 Hardware

Da es in dieser Bachelorarbeit primär um die Entwicklung von Software für Smartphones geht, wird der Bereich der Hardware nur kurz in diesem Abschnitt erläutert.

Einer der großen Vorteile der Multiplattform Fähigkeit des Android Systems ist die Lauffähigkeit auf vielen verschiedenen Hardware Plattformen. So konnten auch günstigere Geräte mit „billiger“ Hardware für eine starke Verbreitung des Systems beitragen.

Abbildung 3 Größenvergleich zwischen dem Samsung Galaxy SII und dem HTC Wildfire

Nachteil der vielen verschiedenen Hardware Plattformen ist der Bedarf an Anpassungsarbeit, der entsteht, wenn Software nicht rein aus skalierbaren Elementen zusammengesetzt ist und somit eine manuelle Feinabstimmung notwendig wird.

Wie aus Abbildung 3 ersichtlich, werden Android Geräte mit verschiedenen Auflösungen gebaut. Dabei gibt es Unterschiede in der Baugröße, sowie in der Auflösung und dem Seitenverhältnis.

Model	Auflösung	Displaygröße
Samsung Galaxy SII	400x800	4,3 Zoll
HTC Wildfire	240x320	3,2 Zoll

Tabelle 2 Vergleich zweier Android Geräte

Ein weiterer Aspekt der beachtet werden muss, ist die Prozessorleistung der Geräte. Hierbei kann bei den Geräten der neueren Generation von einer Mindestleistung von einem Gigahertz ausgegangen werden, was für den Leistungsbedarf der Applikation ausreichend ist.

4.2.2 Sicherheit

Durch das für Android Software genutzte Sandbox System wird stets dafür gesorgt, dass Applikationen nur auf solche Systembereiche zugreifen können, für die ihnen bei der Installation die Erlaubnis gegeben wurde. So kann eine Applikation ohne die entsprechende Erlaubnis nicht auf den Gerätespeicher oder die SD-Karte zugreifen. Auch der Zugriff auf Daten anderer Applikationen ist nur möglich, wenn diese vom selben Entwickler kommen und mit der gleichen Identifikationsnummer signiert wurden. Dies soll verhindern, dass Programme sich unbemerkt Daten der Benutzer aneignen oder personenbezogene Daten ohne Kenntnis des Benutzers verschicken.

5.Softwareentwicklung

In diesem Kapitel geht es um die, während der Bachelorarbeit von mir erstellte, Software, sowie Erläuterungen zum Gebiet der Programmierung für mobile Endgeräte.

5.1 Architektur

Bei der Applikation greifen Elemente der Google eigenen API, mit den von mir selbst erstellten Elementen, wie zum Beispiel für die Animation und der Textanalyse, ineinander.

Beim Start des Programmes werden die grafischen Elemente in den Speicher geladen, sowie die im Layout erstellten Buttons erzeugt. Nach diesem Schritt wird die Basisanimation des Charakters gestartet und der Benutzer hat dann die Möglichkeit mit dem Programm zu interagieren.

Beim Betätigen der Gesprächstaste wird dann, die von Google bereitgestellte Sprachanalyse aufgerufen, die sich vor den laufenden Prozess der Applikation legt. Nach dem Aufzeichnen der Spracheingabe des Benutzers wird diese komprimiert und an den Google Server weitergeleitet. Dieser gibt dann den erkannten Inhalt an die Applikation zurück.

Die Applikation kann im Anschluss daran den erkannten Inhalt nutzen um den Antwortsatz zu generieren.

Sobald dieser Antwortsatz erzeugt wurde, wird er über Text-to-Speech wieder in eine hörbare Sprache umgewandelt und wiedergegeben. Die Text-to-Speech Datenbank befindet sich direkt auf den Geräten, es wird hierfür keine weitere Internetverbindung benötigt.

Sobald dieser Vorgang abgeschlossen ist, kann der Benutzer sich den Satz wiederholen lassen oder eine neue Eingabe tätigen.

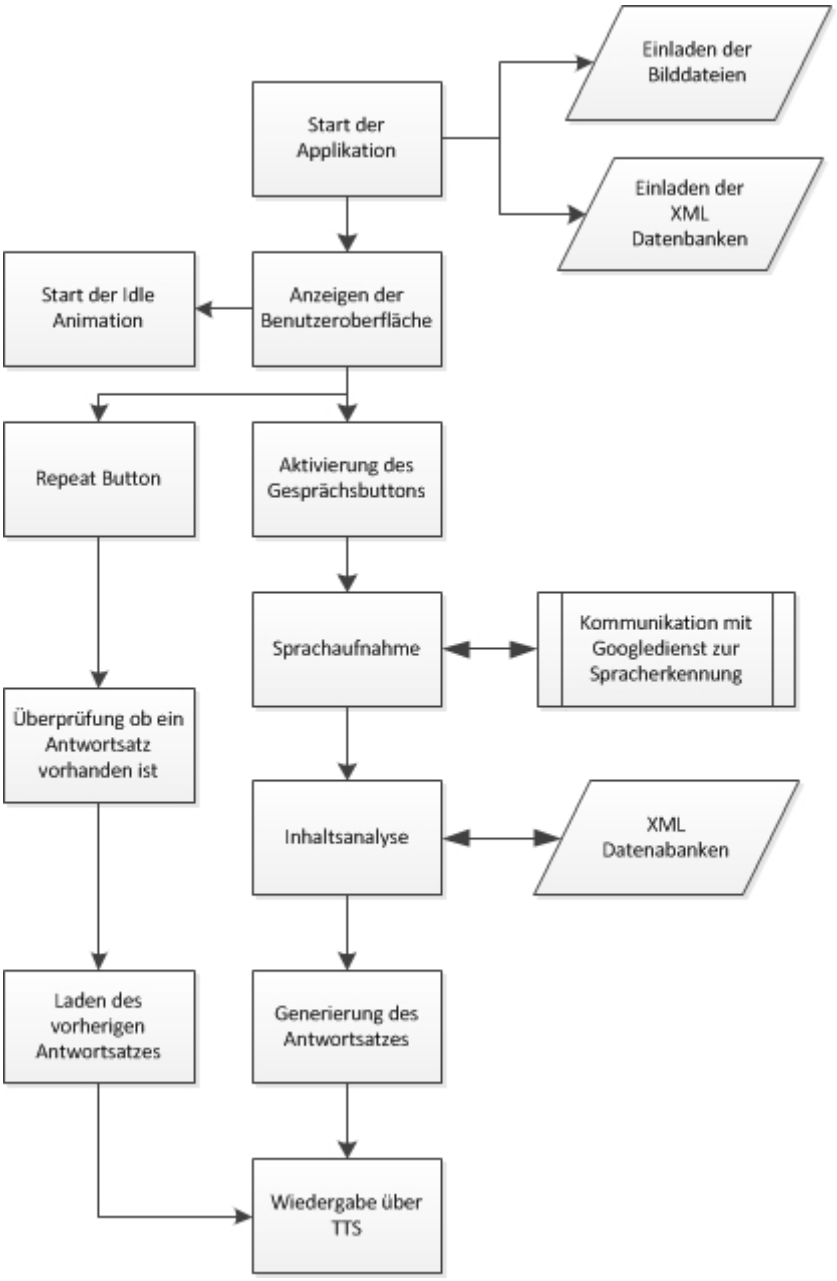

Abbildung 4 Verlaufsdiagramm der Applikation

5.2 Konzept

Im Folgenden Abschnitt werden besonders relevanten Vorgaben und Bereiche erläutert, die durch die Wahl des Android Betriebssystems entstehen.

5.2.1 Displays und Auflösung

Smartphones und Tablets mit installiertem Android Betriebssystem haben zurzeit eine Auflösung von 240x320 bis zu 720x1280 Pixeln. Da eine Lauffähigkeit auf allen Smartphones sowie Tablets gewährleistet werden soll muss beim Layout und auch bei der Auswahl der wiederzugebenden Bilder auf eine ausreichend hohe Auflösung geachtet werden.

Möglichkeit der Gewährleistung der passenden Darstellung auf allen Geräten wäre einerseits die Skalierung aller Bilder während der Laufzeit direkt auf dem Gerät. Dies kann jedoch, da es sich nicht immer um das gleiche Seitenverhältnis handelt, zu einer Verzerrung des Bildes führen.

Andererseits kann dem Problem durch das Hinterlegen der Grafiken in verschiedenen Auflösungen und Seitenverhältnissen entgegengewirkt werden. Dies würde jedoch dazu führen, dass die benötigte Speichermenge der Applikation steigen würde.

Eine Lösung hierfür ist das Hinterlegen der Grafiken in bis zu drei Qualitätsstufen. So können die Grafiken, wenn sie nicht genau passen, Notfalls mit einem schwarzen Rahmen dargestellt werden.

Abbildung 5 Größenvergleich der Animation bei 3 verschiedenen Qualitätsstufen

Die Auswahl der Qualitätsstufen und Skalierung der Bilder wurden hierbei im Zuge der Bachelorarbeit selbst vorgenommen.

5.3 Bedienung

Die Idee des Bedienungskonzepts dieser Applikation ist ein intuitives und übersichtliches Layout. Hierbei wird auf unnötige Oberflächenelemente verzichtet. Die Bedienung erfolgt über einen Eingabeknopf der die Sprachaufnahme aktiviert.

Abbildung 6 Screenshot aus der laufenden Applikation mit Layoutelementen

Sobald die Sprachaufnahme aktiviert ist, beginnt die Aufzeichnung der Spracheingabe des Benutzers. Diese wird im Anschluss an den Google Service zur Sprachanalyse übertragen, der dann einen entsprechenden String zurückgibt.

Nach der Rückgabe der Daten durch den Google Service wird der Inhalt der Spracheingabe des Benutzers analysiert.

Daraufhin wird die Antwort generiert und wieder in Sprache umgewandelt. Diese wird dann mit einer zusätzlichen optischen Aufbereitung dem Benutzer wiedergegeben.

Sollte der Benutzer die Lautstärke des Gerätes zu sehr reduziert haben oder sollte er abgelenkt gewesen sein, kann er über einen entsprechenden Button die generierte Antwort erneut anhören.

Um eine Bedienbarkeit auf allen Geräten gleichermaßen gewährleisten zu können, werden relative Layouts verwendet, die für eine automatische Skalierung sowie Positionierung der Oberflächenelemente sorgen. So können Menü und Steuerungselemente unabhängig von der Displaygröße oder Auflösung positioniert werden und es bedarf keiner individuellen Positionierung der Bedienelemente abhängig vom Gerät.

5.4 Dateistruktur

Bei der Entwicklung der Applikation muss sich streng an die durch das Android Software Development Kit (SDK) vorgegebenen Regeln gehalten werden. Dies wird hier beispielhaft an der während der Bachelorarbeit entwickelten Applikation erklärt. (BP10)

Pflichtbestandteil einer jeden Applikation sind folgende Dateien:

- [Programmname].java
- AndroidManifest.xml
- Layout.xml
- R.Java

5.4.1 Java Code

In der [Programmname].java wird der vom Benutzer entwickelte Code gespeichert. Genutzt werden können im Grunde alle Befehle, die auch aus der Desktop Java Version bekannt sind. Hinzu kommen jedoch noch Smartphone spezifische Klassen wie zum Beispiel Aufrufe an Sensoren oder Abfragen von Eingaben auf dem Touchscreen. Die Anzahl der Smartphone spezifischen Klassen hängt stark von der verwendeten Version des SDK ab.

Beim SDK handelt es sich um die Version der Entwicklungsumgebung. Da sich dieses, durch den Hersteller, in einer permanenten Weiterentwicklung befindet, ändern sich infolge dessen auch die verfügbaren Klassen und Methoden, die in der

jeweils aktuellen Version genutzt werden können. So ist zum Beispiel erst ab der Android Version 3.0, was der SDK Version 11 entspricht, die Möglichkeit der Hardwarebeschleunigung von Grafiken gegeben.

Bei der Anforderung, dass die Software auch auf älteren Geräten lauffähig sein soll, kann eine ältere SDK Version genutzt werden, die das Nutzen neuer Methoden vollständig unterbindet. Eine weitere Möglichkeit wäre es im Code sicherzustellen, dass die entsprechend neueren Methoden nur auf den Geräten benutzt werden, die über die benötigten Anforderungen verfügen.

Versucht wird hierbei eine Lauffähigkeit auf Android Geräten mit der Version 1.6 zu gewährleisten, da so auch Geräte der früheren Android Generationen die Applikation nutzen können.

5.4.2 AndroidManifest.xml

In der AndroidManifest.xml werden grundlegende Informationen zur Applikation gespeichert. Als erstes werden die Versionsnummer sowie der Versionscode festgelegt. Bei der Versionsnummer handelt es sich um eine frei festlegbare Ziffernfolge, wie zum Beispiel Version 3.22 . Der Versionscode jedoch muss bei jedem Erstellen einer neuen Version für den Google Play Store angehoben werden und „zählt“ also mit den entwickelten Versionen immer um eine volle Zahl hoch. Des Weiteren werden im Manifest der Applikationsname sowie das Icon festgelegt. Auch Berechtigungen, wie zum Beispiel der Zugriff auf Daten der SD Karte oder der Internetzugriff müssen hier festgelegt werden. Dabei handelt es sich um essenzielle Freigaben. Wird zum Beispiel im Android Manifest die Freigabe für die Nutzung des Internets nicht gegeben, kann dieses auch nicht verwendet werden. Dies dient primär dem Schutz des Endnutzers. Dieser kann bei der Installation der Applikation direkt sehen, welche Rechte der Applikation eingeräumt werden und kann dann entscheiden ob er diese einräumen möchte.

Soll in der Applikation auch die Möglichkeit bestehen andere Applikationen oder deren Bestandteile aufzurufen, muss dieses im Manifest ebenfalls festgelegt werden. Dies wird zum Beispiel genutzt um Werbung in die Applikation zu integrieren.

5.4.3 Layout.xml

Bei der Layout.xml handelt es sich um eine Datei, die es erlaubt in XML Form ein Layout zu erstellen, das sich dann auch an verschiedene Displaygrößen anpassen lässt. In diesem können, mit dem in Eclipse integriertem XML Builder per Drag and Drop Verfahren, einfache Layouts erstellt werden.

Die im Layout festgelegten Elemente wie Buttons, Textfelder und andere Objekte können dann im Programmcode direkt angesprochen werden, um ihnen Funktionen zuzuweisen oder sie zu bearbeiten.

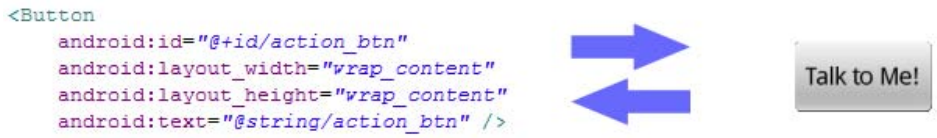

Abbildung 7 Darstellung eines Buttons und seine zugehörige Definition in der Layout.xml

5.4.4 R.Java

Bei der R.Java handelt es sich um eine automatisch erstellte Index Datei. In dieser, durch Eclipse erstellten, Datei wird bei der Kompilierung jedem Objekt eine Identifikationsnummer zugewiesen, die es eindeutig bestimmt. Über diese Identifikationsnummer kann das Objekt dann später im Programm aufgerufen und identifiziert werden. Bei der Zuweisung wird dem jeweiligen Namen des Objektes ein Integer Wert zugewiesen. Dabei wird als Variablenname der Name des Objektes verwendet und diesem die Identifikationsnummer zugewiesen. Ein Aufruf eines Objektes erfolgt innerhalb der Applikation dann über einen Aufruf aus dieser Datei. In dieser Index Datei werden jedoch keine zur Laufzeit erstellten Objekte gespeichert, sondern nur Layoutelemente sowie Bilder und Sound Dateien.

5.5 Ordnerstruktur

Im folgenden Abschnitt geht es um die Ordner- und Programmstruktur der Applikation. Jeder Applikation liegt eine gewisse Ordner und Dateistruktur zu Grunde, von der nicht abgewichen werden darf.

5.5.1 SRC Ordner

Im SRC Ordner befindet sich der eigentliche Programmcode der Applikation. Der Aufbau einer Android Applikation ist der einer Java Anwendung, vor allem im Bereich des Java Programmcodes, sehr ähnlich. Es werden einzelne Pakete angelegt, in denen sich dann die entsprechenden Java Dateien befinden.

5.5.2 gen Ordner

Im gen Ordner befindet sich eine durch Eclipse automatisch generierte Datei. Bei dieser Datei handelt es sich um die R.Java. In dieser Datei wird, wie bereits weiter oben ausführlich erklärt, eine Identifikationskartei erstellt.

5.5.3 lib Ordner

Im lib Ordner werden importierte Bibliotheken abgelegt, die dann im Programm verwendet werden können. Dabei kann es sich um andere Android Programme handeln oder auch um externe Bibliotheken die zum Beispiel für die Einbindung von Werbung in die Applikation genutzt werden.

5.5.4 res Ordner

Im res Ordner befinden sich alle für die Applikation genutzten Ressourcen, wie Bilder, Sounddateien, Layouts und Daten in Form von XML Dateien.

Um eine Anpassung an verschiedene Geräte zu vereinfachen, können Bilddateien in verschiedenen Auflösungen hinterlegt werden. So greift das Gerät automatisch auf die Dateien zu, die für die entsprechende Pixelauflösung des Gerätes vorgesehen sind. Sollte eine entsprechende Datei in einem der Ordner fehlen, wird diese automatisch aus einem der anderen genutzt und entsprechend skaliert, was dann jedoch zu einem Qualitätsverlust führen kann. Zur Unterscheidung werden hierbei aber nicht die direkten Auflösungen genutzt, sondern die dots-per-inch (dpi), also die Anzahl der Punkte auf einem Inch. Die Unterteilung der Qualitätskategorien erfolgt dabei in 5 Stufen.

Qualität	dots per inch
drawable-ldpi	~120dpi
drawable-mdpi	~160dpi
drawable-hdpi	~240dpi
drawable-xhdpi	~320dpi
drawable-nodpi	-

Tabelle 3 Überblick der verfügbaren Qualitätsstufen

Ein Sonderfall ist hierbei der nodpi Ordner. In diesem können Dateien abgelegt werden, die nicht skaliert werden. Dies dient einer manuellen Auswahl von Bilddateien, um so einen Qualitätsverlust durch Skalierung zu vermeiden.

Sound Dateien werden entweder im wav, ogg oder mp3 Format genutzt und im Unterordner raw abgelegt. Für verschiedene Geräte muss hier keine Anpassung stattfinden, da das jeweilige Soundformat von allen Geräten unterstützt wird. Das Einzige das beachtet werden sollte, ist das eine zu hohe Bitrate zu einer Verzerrung bei der Wiedergabe führen kann.

Die in der Applikation genutzten Layouts werden im Unterordner Layouts abgelegt.

Im Unterordner anim können einfache Animationen, wie sie auch in der Applikation genutzt werden können, im XML Format erzeugt werden. Wie in Abbildung 8 gezeigt wird, wird in der Animationsliste für jedes Bild ein eigenes Item erzeugt, in dem der Speicherort des Bildes sowie die Anzeigedauer des Einzelbildes angegeben wird.

In Abbildung 8 heißen die Bilder der Einzelbilder animation_000X und ihnen ist eine Anzeigedauer von 40 ms zugewiesen. Dies führt bei der Wiedergabe zu einer Bildrate von 25 Bildern pro Sekunde und wird somit als flüssige Animation wahrgenommen.

Die Liste von Einzelbildern kann dann im Programmcode einem Layoutelement zugewiesen werden, das dann die entsprechende Animation abspielt.

```
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">
        <item android:drawable="@drawable/animation_0001" android:duration="40" />
        <item android:drawable="@drawable/animation_0002" android:duration="40" />
        <item android:drawable="@drawable/animation_0003" android:duration="40" />
        .
        .
```

Abbildung 8 Ausschnitt aus der in der Applikation für die Animation genutzten XML Datei

Im Unterordner Values können XML Dateien abgelegt werden, die Texte enthalten. Dies ermöglicht einen Zugriff auf diese von jedem beliebigen Teil des Programmes aus. Standartmäßig befindet sich hier die Datei string.XML, in der der Name der Applikation festgelegt wird.

Für die im Verlauf der Bachelorarbeit erstellte Applikation werden in diesem Bereich weitere Dateien erzeugt, in denen die Daten zur Kommunikation mit dem Benutzer hinterlegt werden. Hinterlegt werden hier die Datenbanken mit Nomen, die erkannt werden sollen sowie die vorgenerierten Antwortsätze.

5.6 Android Spezifische Methoden

Da sich der Funktionsumfang eines mobilen Betriebssystems von dem eines Desktop Computers unterscheidet, müssen bei der Erstellung der Applikation grundlegende Regeln befolgt werden, die für die Lauffähigkeit der Applikation unerlässlich sind. (KH09)

Des Weiteren ist der Zyklus einer Applikation anders als der auf einem Desktop Computer, da bei mobilen Geräten besondere Rücksicht auf den Verbrauch der Batterie genommen werden muss. Zu diesem Zweck werden im Android Betriebssystem zusätzlich Methoden eingefügt die im folgenden Abschnitt weiter erläutert werden.

- onCreate

  Die onCreate Methode wird beim ersten Starten der Applikation aufgerufen. In ihr wird das erste Layout geladen und die Zugriffe auf Buttons und andere Elemente der Benutzeroberfläche initialisiert.

- onStart

  In dieser Methode werden elementare Bestandteile der Applikation gestartet. Hierbei wird die Applikation auf die Oberfläche gelegt und es kann mit ihr interagiert werden.

- onPause/onStop

  Diese beiden Methoden werden aktiv, wenn das Handy in den Standby Modus geht oder der Benutzer die Applikation in den Hintergrund schaltet. In diesen Methoden wird festgelegt, was genau die Applikation in dieser Situation machen soll. Es wird zum Beispiel die Animation des Charakters angehalten, um keine Ressourcen sowie Prozessorleistung unnötig zu verbrauchen.

- onResume

  In dieser Methode wird festgelegt was geschieht, wenn die Applikation wieder aktiv auf die Oberfläche des Handys aufgerufen wird. In ihr sollten also alle für das Programm benötigten Prozesse wieder gestartet werden.

- onDestroy

  Die onDestroy Methode wird aufgerufen, wenn die Applikation beendet wird. In ihr sollten zum Beispiel Elemente wie TTS beendet werden, damit diese nicht nach der Beendigung der Applikation aktiv bleiben. Dies verhindert, dass der Akku des Gerätes auch nach der Beendigung der Applikation weiter durch noch laufende Prozesse belastet wird.

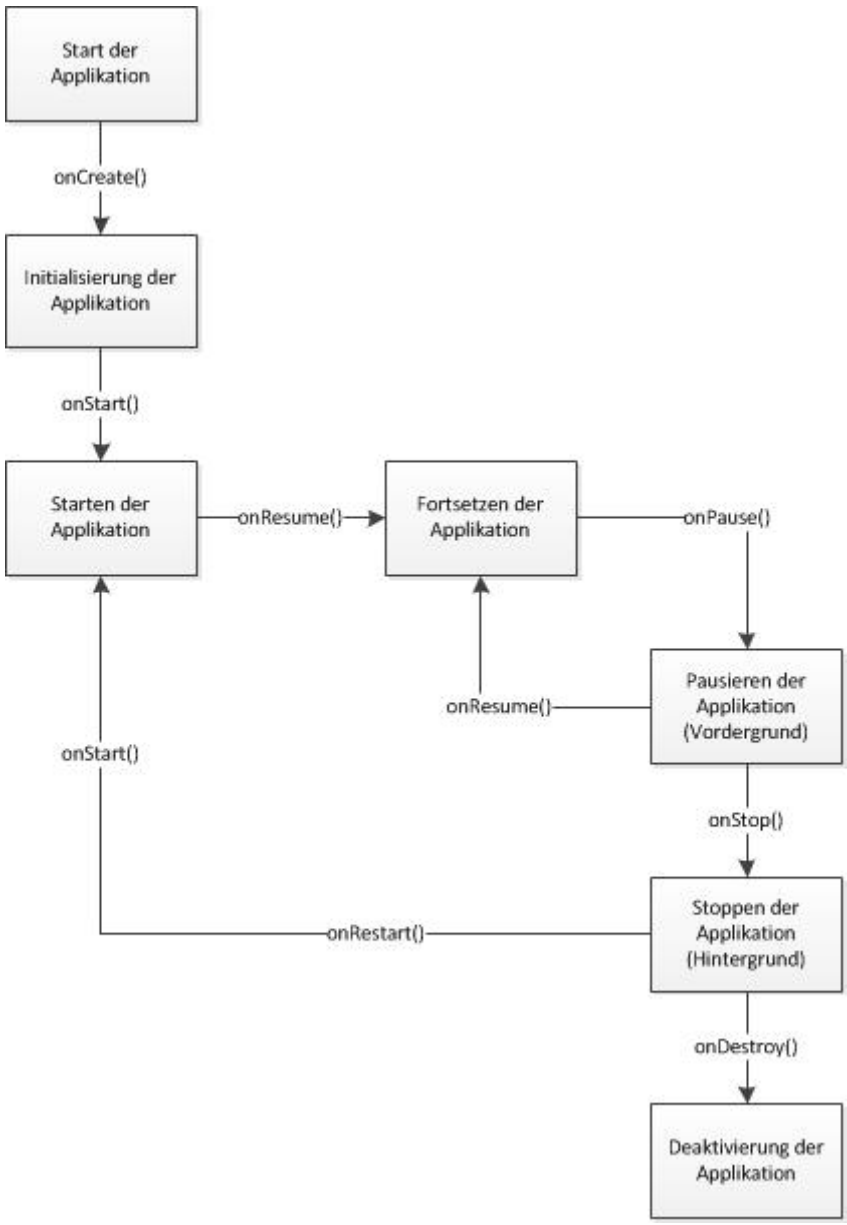

Abbildung 9 Aktivitätszyklus einer Applikation

Neben diesen für die Lauffähigkeit einer Applikation essenziellen Methoden gibt es auch weitere, die die Steuerung der Smartphone spezifischer Hardware ermöglichen.

Darunter fällt zum Beispiel die Nutzung der am Smartphone angebrachten Schultertasten, die im Normalfall für die Steuerung der Klingeltonlautstärke verwendet werden.

Um die Medienlautstärke anpassen zu können, muss den entsprechenden Tasten auf dem Gerät hierbei manuell die entsprechende Funktion zugeweisen werden, da diese sonst auch zweckentfremdet genutzt werden können, zum Beispiel für das Umblättern von Seiten.

Dies geschieht durch das Überschreiben der ursprünglich für die Taste vorgesehenen Methode. Dabei wird zuerst geprüft, welche Taste gedrückt wird. Sollte es sich dann um eine der Lautstärketasten handeln, wird nicht mehr der ursprüngliche Befehl, sondern der im Programm neu gesetzte Befehl ausgeführt.

5.7 Sprachanalyse

Da das ursprüngliche Eliza Programm textbasierend ist, die Eingabe per Hand jedoch auf dem Smartphone viel Zeit kosten würde, musste eine andere Lösung gefunden werden. Dabei lag das Augenmerk vor allem auf der Usability für den Benutzer. Für diesen sollte die Eingabe möglichst intuitiv, einfach, sowie schnell sein. Die einfachste und auch technisch sinnvollste Lösung war hierbei die Nutzung der Sprachanalyse, auch STT (Speech-to-Text) genannt. Bei dieser von Google breitgestellten Funktion, wird während des Sprechens ein Audiofile aufgenommen und dann in komprimierter Form über das Mobilfunknetz bzw. W-LAN an den Google Server übertragen. Auf diesem wird der Audiofile analysiert und das Ergebnis wird dann in Form eines Strings an das Endgerät zurückgeschickt.

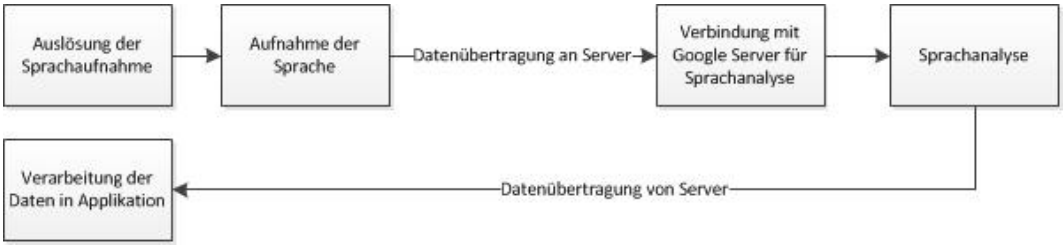

Abbildung 10 Datenweg einer Sprachaufnahme

Die Nutzung dieser Struktur erlaubt es, ohne Einsatz eigener Server, ein sicheres und verlässliches System der Sprachanalyse zu nutzen.

Ein Vorteil der Nutzung der Google Strukturen ist es, dass diese permanent von Google verbessert werden. So kann die Applikation bei jedem Update der Sprachanalyse von diesem profitieren, ohne dass eine weitere Modifikation des Codes nötig wäre.

5.8 Textanalyse und Antworterstellung

Bei der Textanalyse handelt es sich um das eigentlich Kernstück der Anwendung. Um überhaupt eine sinnvolle Konversation mit dem Anwender ermöglichen zu können, ist es von Nöten, die sprachlichen Eingaben richtig zu interpretieren und zu analysieren.

Für diesen Zweck wurden entsprechende Schlüsselwörter in einer XML Datenbank gespeichert. Wenn eines oder mehrere dieser Wörter sich im Eingabetext befinden, kann daraus ein entsprechender Antwortsatz generiert werden.

Die Struktur der XML Datenbank ist dabei wie folgt aufgeteilt. Es wurde für die jeweilige Wörtergruppe (Nomen) bzw. Schlüsselsätze ein String Array erstellt, mit dem dann der Eingabetext sequentiell verglichen werden kann. Des Weiteren wurden String Arrays erstellt, die mit häufig genutzten Inhalten gefüllt sind, auf die dann ein individueller Antwortsatz gegeben wird. Bei der normalen Generierung des Antwortsatzes werden Wörter aus dem Eingabetext ausgelesen, mit dem Array verglichen und anschließend in einen von mehreren vorgefertigten Antwortsätzen eingepasst.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="nouns">
        <item>exam</item>
        <item>university</item>
        <item>lesson</item>
    .
    .
    .
    </string-array>
</resources>
```

Abbildung 11 Ausschnitt aus der XML Datei mit Nomen

Bei der Erstellung des Antwortsatzes wird im Detail wie folgt vorgegangen.

Der durch die STT Schnittstelle erstellte String, mit der Eingabe des Benutzers, wird im ersten Schritt in einem String Array gespeichert. In diesem wird jedes einzelne Wort, ohne etwaige Leer- oder Sonderzeichen (A, Ö, Ü, ß), als eigener String abgelegt. Des Weiteren wird ein Array erstellt, in dem gespeichert werden kann, ob ein Wort einerseits vorkommt und andererseits an welcher Position es sich in der XML Datenbank befindet.

Daraufhin werden die einzelnen Wörter des Eingabesatzes mit den in der XML Datenbank vorhandenen Wörtern verglichen und festgestellt, welche Wörter sowohl in dem Eingabesatz, als auch in der XML Datenbank vorkommen. Dabei wird auch geprüft, ob es sich nur um ein einzelnes erkanntes Wort handelt oder ob eine ganze Aussage wie zum Beispiel *„Auf Wiedersehen“* erkannt wurde. Diese werden zwischengespeichert, um später für den Antwortsatz genutzt werden zu können.

Danach wird auf Grund dieses Arrays analysiert, wie viele Wörter sich in dem Eingabestring befinden, die es bei der Erstellung des Antwortsatzes zu berücksichtigen gibt und um welche Art es sich bei diesen handelt.

Aus den so gewonnenen Daten (Anzahl und Art der Worte) lässt sich nun ein Antwortsatz generieren. Hierfür sind in einer weiteren XML Datenbank Satzfragmente gespeichert. Diese bestehen jeweils aus einem Satzanfang und dem Satzende. Sie lassen sich, zusammen mit den aus dem Eingabesatz entnommenen Worten, zu vollständigen Sätzen kombinieren. Dieser dann zusammengestellte String kann dem Benutzer wiedergegeben werden.

Sollte sich keines der Schlüsselwörter in dem Eingabetext befinden, wird das Gespräch automatisch mehr in die gewünschte Richtung gelenkt bzw. der Anwender gebeten das Thema zu wechseln.

Wurde jedoch im vorherigen Durchlauf eines Zyklus bereits ein Wort erkannt, wird dieses genutzt, um das Gespräch wieder in diese Richtung zurückzulenken, indem der Antwortsatz sich noch einmal auf das vorherige Wort bezieht.

5.9 Anwendungsbeispiele

Bei der Nutzung der Applikation wird der Benutzer als erstes durch den Charakter begrüßt. Es findet keine Erklärung in Text- oder Tonform statt. Der Benutzer soll durch ausprobieren und interagieren mit dem Charakter selbst begreifen, wie die Applikation funktioniert und was ihre Funktion ist.

Dies geschieht im ersten Schritt durch die Begrüßung durch den Charakter. Der Benutzer merkt so direkt, dass eine Interaktion möglich ist. Daraufhin wird auf die Aktivierung der Spracherkennung durch den Benutzer gewartet. Sobald dieser einen Eingabesatz in das Gerät gesprochen hat, wird er analysiert und ein Antwortsatz generiert.

Diese Generierung unterteilt sich in 3 Fälle:

- Fall 1: Im Eingabesatz wird ein Wort erkannt.

Bei einer Übereinstimmung eines Wortes aus dem Eingabesatz mit den in der XML Datenbank gespeicherten Wörtern, wird ein zufällig ausgewählter Antwortsatz aus der XML Datenbank, die die Antwortsätze beinhaltet, entnommen und mit dem Eingangswort kombiniert. Dabei werden die Einzelteile des Antwortsatzes mit dem gefundenen Wort zu einem vollständigen Satz verbunden.

Eingabe : „Ich war heute im Kino.“

Die Applikation erkennt hier beispielhaft das Wort Kino und nutzt dieses um einen Antwortsatz zu generieren.

Ausgabe : „Erzähl mir mehr von dem Kino das du erwähnt hast.“

Dabei handelt es sich bei den Satzfragmenten *„Erzähl mir mehr von dem „* und *„das du erwähnt hast“* um die vorbereiteten und in der Applikation hinterlegten Bruchstücke der Antwortsätze.

- Fall 2: Im Eingabesatz wird kein Wort erkannt.

Sollte sich im Eingabesatz kein Wort befinden, das der Applikation bekannt ist, wird diese auf zwei, den entsprechenden Umständen angepasster, verschiedener Art reagieren. Sollte sie in der vorherigen Eingabe ein Wort erkannt haben, wird sie das letzte gefundene Wort in einem dafür vorbereiteten Satz nochmal nutzen, um das Gespräch wieder in einen Bereich zu bewegen, der in der Datenbank enthalten ist.

Eingabe : „Der Vorhang ging nicht auf.“

Die Applikation erkennt hier das Wort Vorhang nicht und reagiert darauf, indem es das zuvor erkannte Wort noch einmal nutzt. Der Antwortsatz wird hierbei wie bereits in Fall 1 erläutert erzeugt.

Ausgabe : „Lass uns doch über etwas anderes im Kino sprechen.“

Sollte es sich jedoch um die erste Eingabe des Benutzers handeln und es wird kein Wort erkannt, wird, durch für diesen Fall vorbereitete Sätze, der Benutzer in Themengebiete gelenkt für die in der Applikation Inhalt bereit gestellt ist.

Ausgabe : „Lass uns doch über deinen Tag reden.“

- Fall 3: Standard Aussage gefunden.

Sollte die Eingabe eine häufig gebrauchte Sprachwendung wie beispielweise eine Begrüßung oder eine Verabschiedung beinhalten, wird auf diese durch eine fest vorgegebene Antwort reagiert.

Eingabe : „Auf Wiedersehen.“

Ausgabe : „Hoffentlich sehen wir uns bald wieder.“

Wenn der Benutzer keine Interaktion mit dem Charakter tätigt, bewegt sich dieser um eine Lebendigkeit zu suggerieren.

5.10 Produktentwicklung

Bei der Produktentwicklung wurden nach eingehender Recherche, welche Möglichkeiten das Android System an sich bietet, von mir entsprechende Klassen und Funktionen herausgesucht und auf ihre jeweilige Tauglichkeit überprüft.

Ein anderer Teil der Produktentwicklung waren die Kommunikation und Meetings innerhalb der Firma, um nicht nur die eigenen Ideen und Konzepte einzubringen, sondern auch in Rücksprache mit Kollegen und Projektleitern zu treten.

Daraufhin folgte meinerseits das Erstellen der, für die Nutzung auf dem Smartphone angepassten, Textanalyse, das Layout sowie die Erzeugung von benötigten Methoden.

Nach Abschließen dieser Schritte folgte das Kombinieren aller geprüften und selbst erstellten Programmteile zu einer Gesamtanwendung.

5.11 Technische Daten

Grundvoraussetzung für die Nutzung der Applikation ist eine installierte Android Installation auf dem Endgerät mit der Mindestversion 1.6 .

6.Charakterdesign

Da es in Applikationen heutzutage üblich ist, nicht nur eine sprachliche oder textbasierende Antwort auf seine Interaktion zu erhalten, sondern auch eine optische, wurde auch dieses Element in die Applikation eingebettet.

6.1 Auswahl des Charakters

Die Auswahl des eingesetzten Charakters geschah in Zusammenarbeit mit dem Projektleiter. Da die Applikation im Anschluss auch vertrieben werden soll, mussten in diesem Bereich Absprachen gemacht werden, um die spätere Verwertbarkeit zu garantieren.

Geeinigt wurde sich auf einen Papageien, da dieser einerseits von Natur aus in den Bereich der Sprachsynthese passt und andererseits auch die entsprechende Vermarktbarkeit gewährleistet.

6.2 Entwicklung und Animationen

Für die Entwicklung des Charakters und dessen Animation wurde auf die Arbeit eines Professionellen 3D-Designers zurückgegriffen. Dieser erstellte in Absprache mit mir mehrere Animationen, auf die dann in der Applikation zurückgegriffen werden konnte.

Dazu gehören einerseits Sprachanimationen, sowie mehrere Animationen, wenn der Benutzer keine Eingabe tätigt. Dies vermittelt dem Benutzer eine Art Lebendigkeit des Charakters mit dem er interagiert.

Abbildung 12 Der in der Applikation verwendete Charakter

7.Zusammenfassung

Die Aufgaben während der Bachelorarbeit teilten sich in zwei Gebiete auf. Einerseits ging es um das Erstellen der Software unter Berücksichtigung der systemspezifischen Eigenheiten. Andererseits musste das System hinter der Sprachsteuerung und die dabei verwendeten Methoden integriert werden. Diese beiden Elemente mussten dann miteinander verknüpft werden, um ein stimmiges Gesamtprodukt zu erzeugen. Dabei war auch die Wahl des Android Systems als Plattform korrekt. Diese ermöglichte das Nutzen der im Studium erlangten Erfahrungen, in Kombination mit den während des Industriepraktikums erlangten Fachkenntnissen.

Während der Arbeit wurden bereits bekannte Gedankengänge und Ideen wie das Eliza Programm mit den momentan aktuellen technischen Gegebenheiten verknüpft, um so ein neues Produkt zu erschaffen.

Im Verlauf dieser Bachelorarbeit wurden Einblicke in Gebiete der Softwareentwicklung vor allem im Bereich der mobilen Endgeräte, insbesondere der Android Plattform, gegeben. Wobei sowohl die Hardware hinter dem System sowie auch die darauf laufende Software erläutert wurden. Infolge dessen konnte dann ein Überblick über die Systemstrukturen und den Anpassungsbedarf des Quellcodes getroffen werden.

Der während der Bachelorarbeit erstellte Quellcode wird auch anderen Programmierern der Firma *Nurogames* zur Verfügung gestellt, um diesen in bereits existierende Produkte einzuflechten und so diese um weitere Aspekte zu erweitern.

8.Ausblick

Da die Software für die Veröffentlichung und den Verkauf vorgesehen ist, muss auf die Erweiterbarkeit geachtet werden. Des Weiteren muss gewährleistet sein, dass sich die Software in bereits vorhandene Programme integrieren lässt, um diese mit der Funktion der Textanalyse zu erweitern. Dies wurde während der Erstellung der Software bereits bedacht und wird in den nun folgenden Punkten erläutert.

8.1 Benutzeroberfläche

Da mein Augenmerk bei der Entwicklung der Software mehr auf der technischen Seite, sowie der Textanalyse liegt und eine komplette gestalterische Ausarbeitung den Rahmen dieser Bachelorarbeit überschreiten würde, bleiben in diesem Bereich noch Möglichkeiten der Erweiterung von professionellen Designern und Grafikern.

8.2 Charakter Erweiterbarkeit

Da in der entwickelten Version die Darstellung des Charakters auf Einzelbildern beruht, die bereits bei der Installation mitgeliefert werden, können nur diese als Ressourcen genutzt werden.

Eine andere Möglichkeit der Erzeugung der optischen Benutzeroberfläche wäre das Streaming von 3D Modellen direkt von einem Server. Hierbei könnte man sogar die Lippensynchronität während der optischen Darstellung des Charakters gewährleisten. Dies lässt sich jedoch erst sinnvoll umsetzen, wenn eine flächendeckende mobile Datenüberragung mit entsprechenden Kapazitäten und Geschwindigkeiten gewährleistet ist.

8.3 Spracherkennung

Bei der Spracherkennung wird die Google eigene Speech-to-Text API verwendet. Dies führt dazu, dass eine permanente Internetverbindung benötigt wird, um die Applikation zu verwenden. Verbessern lässt sich dies entweder über die Entwicklung einer eigenen Spracherkennung, die auch ohne eine Internetverbindung funktioniert, oder durch einen Ankauf einer Fremdsoftware, die ein entsprechendes Feature besitzt.

8.4 Sprachwiedergabe

Wie auch bei der Spracherkennung wird die Google eigene API für die Umwandlung von Text in Sprache verwendet. Da diese Aussprache sehr mechanisch klingt, kann man auch in diesem Bereich durch die Erstellung eigener Sprachdatenbanken eine Verbesserung der Gesamtsoftware erreichen.

8.5 Erweiterung der Analysefaktoren

Durch die in Android 4.0 eingeführte Gesichtserkennung lassen sich später eventuell auch emotionale Ausdrücke aus dem Gesicht des Anwenders auslesen und in die Analyse einflechten. Auch die Analyse der eingesprochenen Worte im Bezug auf Tonhöhe und Frequenz bei gleichzeitiger Zuordnung zu einem emotionalen Zustand lassen noch Raum für Erweiterungen.

8.6 Automaische Verbesserungen

Durch das Nutzen der Google eigenen Text-to-Speech und Speech-to-Text API’s wird die Applikation ohne das Zutun des Entwicklers stetig verbessert, da jedes Update der Google Sprachsteuerung automatisch auch im Programm genutzt wird.

Quellenverzeichnis

[Rog93] Carl R. Rogers: *Die klientenzentrierte Gesprächspsychotherapie*. Frankfurt a. M., Deutschland : Fischer TB, 1993

[ERB09] Robert Epstein; Gary Roberts; Grace Beber: *Parsing the Turing Test.* Niederlande : Springer Science, 2009

[BP10] Arno Becker; Marcus Pant: *Android 2. Grundlagen und Programmierung*. Deutschland : dpunkt.verlag, 2010

[KH09] Satya Komatineni; Sayed Hashimi: *Pro Android.* Apress, 2009

[Wei78] Joseph Weizenbaum: *Die Macht der Computer und die Ohnmacht der Vernunft*. Frankfurt, Suhrkamp, 1978

Abbildungsverzeichnis

Tabellenverzeichnis

Eidesstattliche Erklärung

Ich versichere hiermit, die vorgelegte Arbeit in dem gemeldeten Zeitraum ohne fremde Hilfe verfasst und mich keiner anderen als der angegebenen Hilfsmittel und Quellen bedient zu haben.
Köln, den 28. Juni 2012

Unterschrift

(Rouven von der Burg)